

POLITECHNIKA ŚLĄSKA W GLIWICACH

ALGORYTMY KOMPRESJI DANYCH  
18 STYCZNIA 2017

---

# Porównanie PNG, JPEG-LS i JPEG2000 w kompresji bezstratnej

---

AUTOR:  
Bartłomiej Buchała

Informatyka SSM, semestr II  
Rok akademicki 2016/2017  
Grupa OS1

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Użyte algorytmy</b>	<b>2</b>
2.1	PNG . . . . .	2
2.2	JPEG-LS . . . . .	2
2.3	JPEG2000 . . . . .	3
<b>3</b>	<b>Wykorzystane biblioteki</b>	<b>3</b>
3.1	PNG – pnmtpng . . . . .	3
3.1.1	Instalacja . . . . .	3
3.1.2	Uruchamianie . . . . .	3
3.2	JPEG-LS – SPMG/JPEG-LS . . . . .	4
3.2.1	Instalacja . . . . .	4
3.2.2	Uruchamianie . . . . .	4
3.3	JPEG2000 – JasPer . . . . .	4
3.3.1	Instalacja . . . . .	4
3.3.2	Uruchamianie . . . . .	5
<b>4</b>	<b>Porównanie wyników</b>	<b>5</b>
4.1	Platforma testowa . . . . .	5
4.2	Badane wartości . . . . .	5
4.3	Obrazy barwne . . . . .	5
4.4	Obrazy czarno-białe . . . . .	9
<b>5</b>	<b>Wnioski</b>	<b>12</b>

## 1. Wstęp

Cel projektu: przeprowadzić porównanie algorytmów PNG, JPEG-LS i JPEG2000 w trybie kompresji bezstratnej. Porównywane implementacje ocenić pod względem:

- Uzyskiwanych współczynników,
- Prędkości kompresji.

Należy przeprowadzić badania dla barwnych i czarno-białych obrazów. W tym celu posłużono się obrazami z zestawu Waterloo (<http://links.uwaterloo.ca/Repository.html>):

- Waterloo Greyscale 2 (obrazy w odcieniach szarości)
- Waterloo Colour Set (obrazy barwne)

## 2. Użyte algorytmy

### 2.1 PNG

**PNG** (ang. *Portable Network Graphics*) – to rastrowy format plików graficznych oraz system bezstratnej kompresji obrazów barwnych oraz w odcieniach szarości. Został opracowany w 1995 roku jako następca formatu GIF. Cechy algorytmu PNG:

- Kompresor PNG jest schematem predykcyjnym.
- Kompresja polega na wykonaniu transformacji każdej linii obrazu, po czym skompresowaniu go za pomocą algorytmu deflate.
- Transformacja polega na obliczeniu różnicy pomiędzy wartością piksela, a wartością obliczoną na podstawie funkcji przewidującej.
- Funkcje przewidujące to funkcje wyliczające wartości na podstawie wartości sąsiadujących pikseli.
- Stosowane jest kodowanie LZ77, a do kodowania położenia i długości dopasowanej długości używa się kodów przedrostkowych zmiennej długości (predefiniowanych lub kodów Huffmana z ograniczeniem).

### 2.2 JPEG-LS

**JPEG-LS** jest nowym standardem komitetu JPEG dla kompresji bezstratnej obrazów wynalezionym w 1999 roku. Jest następcą Lossless JPG opartym o algorytm LOCO-I. Wykorzystywany jest ze względu na osiągnięte współczynniki kompresji (bliskie najlepszym) oraz dużą elastyczność umożliwiającą implementację rozszerzeń. Oprócz tego istnieje „prawie bezstratny” (ang. *nearly lossless*) wariant tego algorytmu.

Uogólniony schemat algorytmu JPEG-LS:

1. **Wykrywanie krawędzi pionowych i poziomych** – stosowany w tym celu jest prosty predykt nieliniowy (wyznaczanie jedynie przy pomocy stałopozycyjnych operacji dodawania/odejmowania i porównań).
2. **Kodowanie** – przy pomocy kodera entropijnego i zmodyfikowanej (uproszczonej) rodziny kodów Rice’a. Na tym etapie koduje się również błędy predykcji.
3. **Modelowanie kontekstu** – wyznaczony na podstawie 3 gradientów Model jest dedykowany dla kodów rodziny Rice’a.

## 2.3 JPEG2000

**JPEG-2000** podobnie jak JPEG-LS jest następcą algorytmu JPEG wyłonionym w drodze konkursu. Opracowany w 2000 roku, stanowi standard komitetu JPEG zarówno dla stratnej, jak i bezstratnej kompresji obrazów. Jest algorytmem przeznaczonym dla obrazów barwnych i w odcieniach szarości opartym o transformatę falkową. Pozwala na kodowanie piramidowe i progresywne.

Uogólniony schemat algorytmu JPEG2000:

1. **Srowadzenie nominalnego zakresu jasności pikseli do przedziału symetrycznego względem 0** – jest to krok tożsamy z tym wykonywanym w algorytmie JPEG.
2. **Transformacja przestrzeni barw** – może być nieodwracalna lub odwracalna. Ta pierwsza możliwa jest tylko razem z nieodwracalną transformatą falkową.
3. **Podział składowej na kafelki** – realizowany poprzez nałożenie siatki prostokątnej na obraz (najczęściej o wymiarach 256 na 256 pikseli). Krawędzie siatki nie muszą się pokrywać z krawędziami obrazu, a obraz może mieścić się w jednym kafelku.
4. **Wyznaczenie transformaty falkowej dla każdego kafelka** – przed transformatą należy ekstrapolować wiersze. W przypadku kompresji bezstratnej, stosujemy transformatę odwracalną.
5. **Kwantyzacja współczynników transformaty** – dla kompresji bezstratnej krok zawsze wynosi 1. W przypadku operacji kompresji stratnej koder dobiera krok, aby możliwe było osiągnięcie zadanego współczynnika.
6. **Kodowanie skwantowanych współczynników** – przy pomocy kodera arytmetycznego.
7. **Zapisanie zakodowanych danych w pliku o strukturze opisej przez standard**

## 3. Wykorzystane biblioteki

### 3.1 PNG – pnmtopng

#### 3.1.1 Instalacja

**pnmtopng** jest elementem pakietu NetPBM. Aby umożliwić korzystanie z biblioteki na systemie Unixowym, należy:

1. Posiadać zainstalowane wymagane LibPNG, ZLIB, dowolny kompilator języka C oraz Perl w wersji 6.0 lub nowszy.
2. Pobrać pliki źródłowe spakowane do formatu .tar ze strony SourceForge (<https://sourceforge.net/projects/netpbm/files/>).
3. Wypakować pliki do wybranego przez siebie folderu.
4. Wykonać komendy

```
./configure
make package
./installnetpbm
```

W przypadku Windowsa, należy posłużyć środowiskami Cygwin lub Djgpp.

Prostsza alternatywą jest pobranie skompilowanych plików binarnych ze strony: <http://gnuwin32.sourceforge.net/packages/netpbm.htm>

#### 3.1.2 Uruchamianie

Aby uruchomić kompresję plików, należy posłużyć się aplikacją **pnmtopng**. Do poprawnego działania, w folderze z aplikacją powinny znajdować się również biblioteki *libnetpbm10.dll*, *libpng13.dll* oraz *zlib1.dll*. Przykładowe wywołanie programu w celu kompresji pliku *clegg.ppm*:

```
pnmtopng clegg.ppm >clegg.png
```

## 3.2 JPEG-LS – SPMG/JPEG-LS

### 3.2.1 Instalacja

**JPEG-LS** jest kodekiem implementującym algorytm kompresji i dekompresji do formaty Lossless JPEG (JPEG-LS). Aby móc skorzystać z algorytmu, należy:

1. Pobrać pliki źródłowe ze strony domowej projektu (<http://www.stat.columbia.edu/~jakulin/jpeg-ls/mirror.htm>) i wypakować do wybranego przez siebie folderu.
2. W folderze z wypakowanymi plikami wykonać komendę:

```
| make clean <nazwa>
```

Gdzie <nazwa> określa, który program chcemy skompilować.

- locoe – kompresor
- locod – dekompresor

Po tej czynności utworzą się odpowiednie pliki wykonywalne

### 3.2.2 Uruchamianie

Wywołanie kompresji obrazu można uzyskać za pomocą linii komend:

```
| locoe [dodatkowe_flagi] clegg.ppm [-ooutfile]
```

Nazwa pliku wyjściowego jest opcjonalna, jest ona generowana na podstawie nazwy pliku wejściowego. Alternatywą jest wykorzystanie programu okienkowego **JLSEncoder** dostarczanego razem z plikami źródłowymi. Program ten posiada prosty interfejs pozwalający wybrać pliki z okien dialogowych oraz wyświetla statystyki po wykonaniu konwersji (czas wykonania, stopień konwersji itp.).

## 3.3 JPEG2000 – JasPer

### 3.3.1 Instalacja

**JasPer** jest otwarto źródłową biblioteką zawierającą implementację algorytmu JPEG2000. W celu instalacji na systemie Windows należy:

1. Pobrać i wypakować pliki źródłowe ze strony projektu (<http://www.ece.uvic.ca/~frodo/jasper/>) do wybranej przez siebie lokalizacji.
2. Utworzenie dodatkowy zmiennych środowiskowych:
  - (a) %SOURCE\_DIR% – katalog nadrzędny, w którym wypakowane zostały pobrane pliki
  - (b) %BUILD\_DIR% – ścieżka do katalogu używanego do zbudowania aplikacji
  - (c) %INSTALL\_DIR% – ścieżka do katalogu używanego do zainstalowania aplikacji

Zmienne te są zdefiniowane pliku make i będą wykorzystywane w trakcie instalacji.

3. W wierszy poleceń wykonać polecenie:

```
| cmake -help
```

Pozwala to podejrzeć nazwy wszystkich dostępnych generatorów (programów umożliwiających kompilację plików źródłowych).

4. Za pomocą wybranego przez siebie generatora (parametr -G) wykonać komendę tworzącą plik rozwiązania .sln.

```
| cmake -G "Visual_Studio_14_2015_Win64" -H%SOURCE_DIR% -B%BUILD_DIR% ^ -  
DCMAKE_INSTALL_PREFIX=%INSTALL_DIR%
```

5. W wierszu poleceń programisty (ang. *Developer Command Line*) wykonać komendę:

```
| msbuild %build_dir%\INSTALL.vcxproj
```

Spowoduje to utworzenie skompilowanego programu jasper.exe w ścieżce podanej w %INSTALL\_DIR%.

### 3.3.2 Uruchamianie

Aby uruchomić program `jasper.exe`, należy w pierwszej kolejności skopiować do tego samego katalogu bibliotekę `libjasper.dll`, wygenerowaną wcześniej w folderze `lib`. Może również dojść do sytuacji, w której nie zostanie wykryta biblioteka `ucrtbased.dll`. W takim przypadku należy pobrać ją z internetu i wypakować ją w `C:\Windows\System32\`. Przykładowa komenda uruchamiająca kompresję (plik `clegg.ppm`):

```
| jasper.exe --input clegg.ppm --output clegg.jp2 --output-format jp2
```

## 4. Porównanie wyników

### 4.1 Platforma testowa

Testy przeprowadzono na komputerze stacjonarnym o następującej specyfikacji:

- System operacyjny – Windows 10 Home 64-bit.
- Procesor – Intel Core i5 4590, taktowanie 3.30GHz.
- Pamięć RAM – 8 GB 2-Kanałowy DDR3, taktowanie 1600 MHz.

### 4.2 Badane wartości

Jakość kompresji określono za pomocą współczynnika kompresji, którego można przedstawić przy pomocy wzoru:

$$CR = \frac{SkompresowanyRozmiar}{NieskompresowanyRozmiar}$$

W tym przypadku, im niższa wartość tym lepszy algorytm. Współczynnik można również zapisać procentowo.

Drugą mierzoną wartością był czas kompresji obliczany w milisekundach.

### 4.3 Obrazy barwne

Podstawą dla obrazów barwnych był format **PPM** – odmiana bitmapy, będącej formą zapisu grafiki rastrowej. PPM jest przeznaczony dla obrazów kolorowych i zawiera maksymalnie do 24 bitów na piksel w trybie binarnym (8 bitów na każdy kolor).

Tabela 1: Clegg

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	2099 KB	475 KB	22,62 %	216
		638 KB	30,39 %	58
		1370 KB	65,25 %	273

Tabela 2: Frymire

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	3620 KB	380 KB	10,49 %	176
		914 KB	25,25 %	59
		1561 KB	43,11 %	719

Tabela 3: **Lena3**

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	769 KB	466 KB	60,54 %	103
		436 KB	56,81 %	30
		435 KB	57,93 %	440

Tabela 4: **Monarch**

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	1153 KB	605 KB	52,44 %	230
		543 KB	46,94 %	44
		432 KB	37,44 %	482

Tabela 5: **Peppers3**

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	769 KB	418 KB	54,33 %	149
		377 KB	49,01 %	29
		328 KB	42,64 %	367

Tabela 6: **Sail**

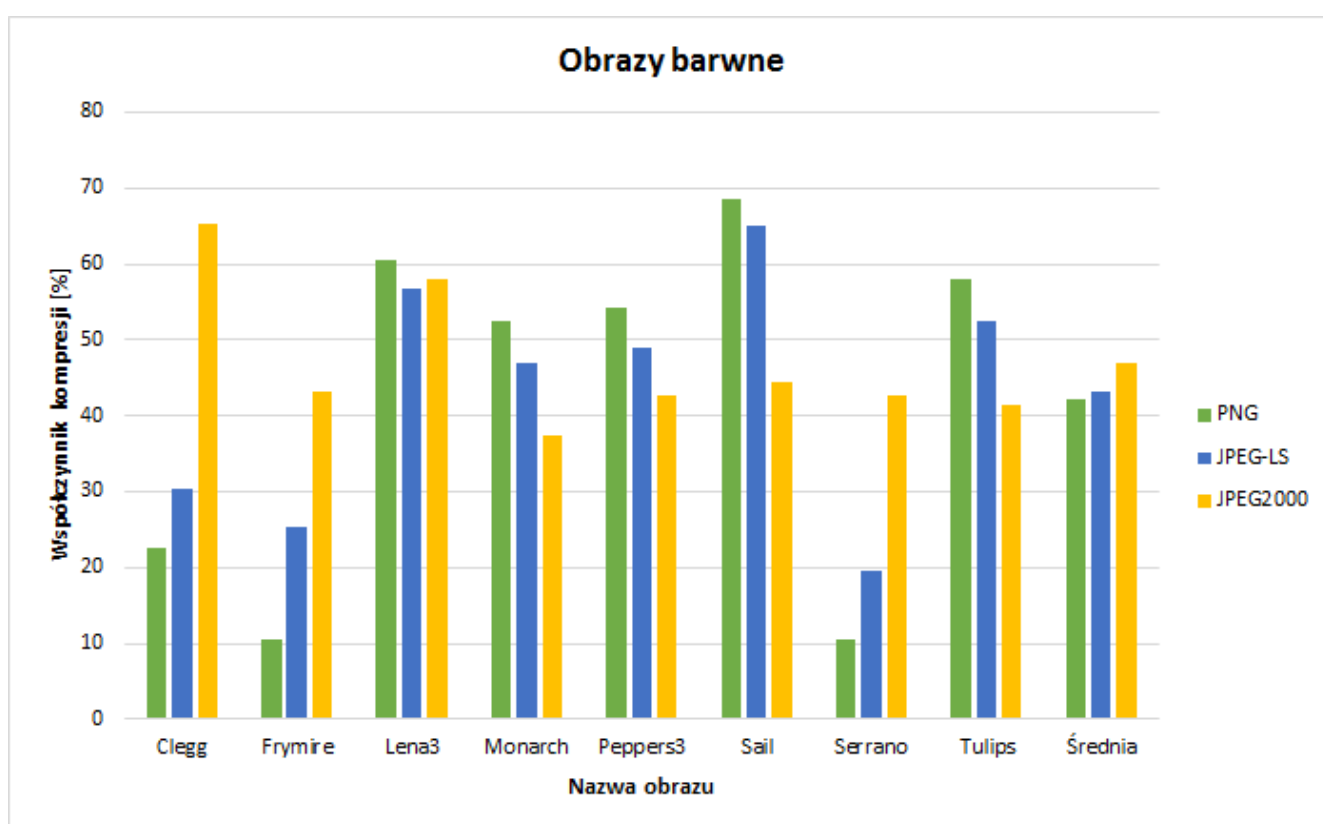
	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	1153 KB	792 KB	68,66 %	156
		750 KB	64,93 %	45
		512 KB	44,37 %	542

Tabela 7: **Serrano**

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	1464 KB	155 KB	10,58 %	95
		287 KB	19,60 %	23
		624 KB	42,61 %	740

Tabela 8: **Tulips**

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	1153 KB	667 KB	57,88 %	202
		603 KB	52,35 %	46
		478 KB	41,46 %	547



Rysunek 1: Wykres słupkowy dla obrazów kolorowych



#### 4.4 Obrazy czarno-białe

Podstawą dla obrazów z zestawu Waterloo Greyset 2 był format **PGM** – odmiana bitmapy, będącej formą zapisu grafiki rastrowej. PGM jest przeznaczony dla w obrazów odcieniach szarości i zawiera 8 bitów na piksel.

Tabela 9: Barb

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	170 KB	66,28 %	58
		152 KB	59,17 %	10
		150 KB	58,24 %	155

Tabela 10: Boat

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	149 KB	57,94 %	67
		136 KB	53,19 %	10
		141 KB	55,07 %	151

Tabela 11: France

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	326 KB	18 KB	5,27 %	40
		58 KB	17,63 %	5
		83 KB	25,25 %	142

Tabela 12: Frog

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	303 KB	227 KB	75,01 %	57
		229 KB	75,75 %	12
		237 KB	78,22 %	125

Tabela 13: Goldhill2

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	157 KB	61,08 %	54
		151 KB	58,82 %	10
		155 KB	60,47 %	171

Tabela 14: Lena2

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	148 KB	57,57 %	63
		136 KB	52,95 %	10
		139 KB	53,95 %	150

Tabela 15: Library

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	160 KB	103 KB	64,18 %	37
		102 KB	63,69 %	6
		114 KB	71,20 %	127

Tabela 16: Mandrill

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	200 KB	76,28 %	49
		194 KB	75,18 %	10
		196 KB	76,37 %	182

Tabela 17: Mountain

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	301 KB	248 KB	82,53 %	65
		240 KB	80,64 %	12
		252 KB	83,76 %	241

Tabela 18: Peppers2

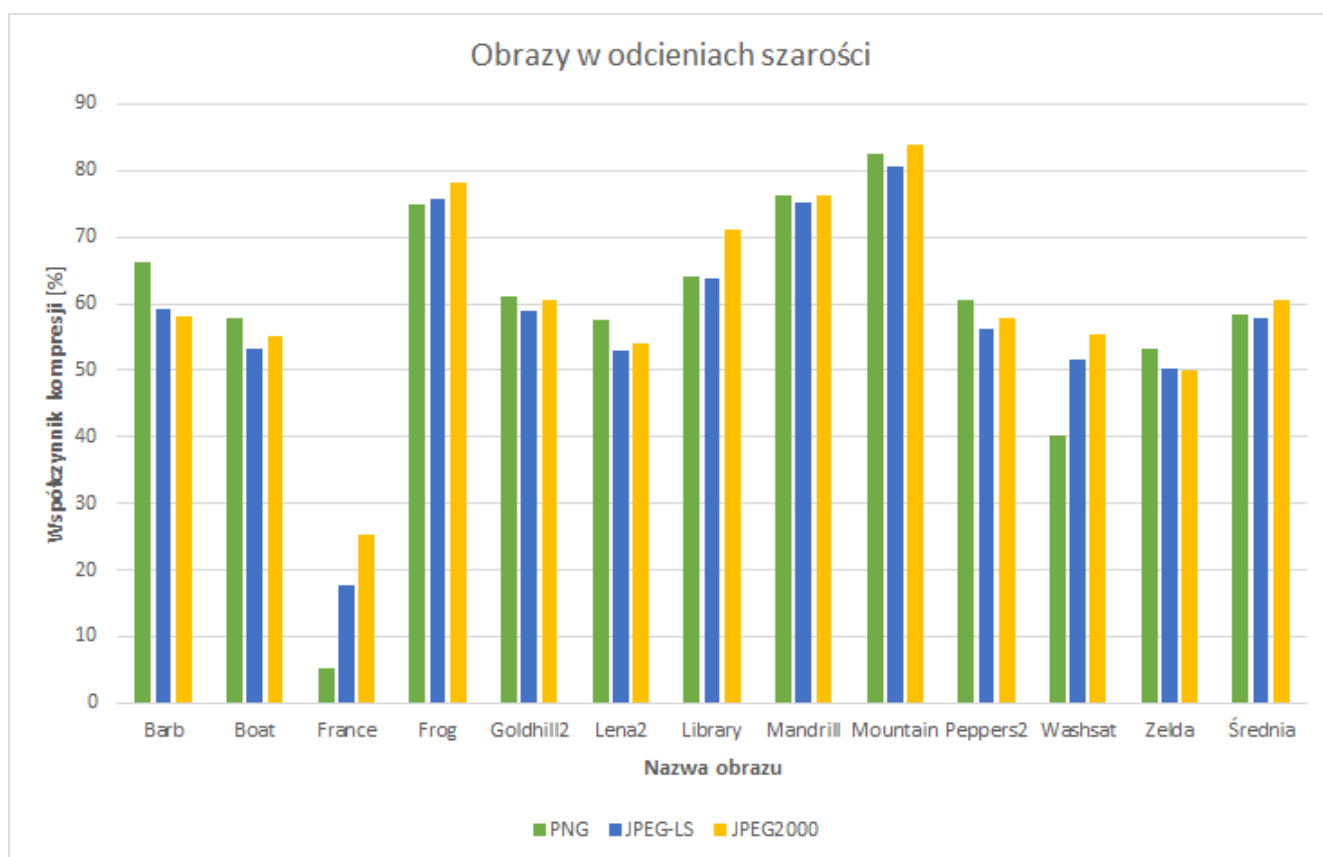
	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	156 KB	60,56 %	56
		144 KB	56,17 %	11
		148 KB	57,73 %	150

Tabela 19: Washsat

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	104 KB	40,35 %	88
		133 KB	51,54 %	9
		142 KB	55,39 %	169

Tabela 20: Zelda

	Rozmiar przed	Rozmiar po	Wspł. kompresji	Czas [ms]
PNG JPEG-LS JPEG2000	257 KB	137 KB	53,17 %	63
		129 KB	50,17 %	9
		128 KB	49,90 %	147



Rysunek 2: Wykres słupkowy dla obrazów w odcieniach szarości

## 5. Wnioski

Wszystkie z testowanych formatów generowane są w trakcie kompresji bezstratnej. Jest to bezpośredni powód, dla którego uzyskane współczynniki kompresji są gorsze niż w przypadku kompresji stratnej. Istnieją jednak zalety korzystania z kompresji bezstratnej – ponieważ nie tracimy danych, są to procesy w pełni odwracalne. W subiektywnym odczuciu różnice między skompresowanymi obrazami nie są widoczne gołym okiem.

Przy porównaniu wyników uzyskanych w trakcie kompresji obrazów barwnych, rzuca się w oczy przede wszystkim różnica w wartości współczynnika dla obrazów *clegg*, *frymire* oraz *serrano*. Powodem tej dysproporcji jest pochodzenie obrazów źródłowych – wszystkie 3 są obrazami stworzonymi cyfrowo. W takim przypadku górę bierze format PNG ze względu na swój algorytm predykcji. W przypadku rysowanych obrazów, częstym scenariuszem jest występowanie w sąsiedztwie pikseli o identycznej wartości, co pozwala PNG na łatwiejsze enkodowanie dużych porcji danych. W takich przypadkach osiągał 3-4 razy lepsze wyniki niż JPEG2000 i 1,5-2 razy lepsze wyniki niż JPEG-LS. W pozostałych przypadkach (*lena3*, *monarch*, *peppers3*, *sail* oraz *tulips*) mamy do czynienia ze zdjęciami – w takim scenariuszach wyniki są bardziej wyrównane, jednak najlepsze współczynniki uzyskiwał algorytm JPEG2000, a najgorzej wypadał PNG. Jedynym wyjątkiem jest obraz *lena3*, dla którego JPEG-LS uzyskał minimalnie lepszy wynik. Przeciętnie najlepiej wypadł format PNG, jednak zawdzięcza to przede wszystkim kompresji obrazów rysowanych.

Nieco inaczej wygląda sytuacja w przypadku obrazów w odcieniach szarości. Znaczące różnice (ponownie na korzyść PNG) można dostrzec głównie dla obrazów *france* i *wahsat*. Pierwszy z nich jest slajdem prezentacji (utworzonym komputerowo), a nieznaczne zmiany w kolorystyce kolejnych pikseli ponownie faworyzuje algorytm PNG. Ciekawszy jest drugi przypadek, przedstawiający krajobraz z lotu ptaka – jest to zdjęcie, jednak charakteryzujące się niewielkim kontrastem pomiędzy pikselami. W pozostałych przypadkach uzyskiwane wyniki były bardzo zbliżone, przeciętnie najlepszy okazywał się format JPEG-LS.

Różnice w szybkości są odczuwalne (zwłaszcza dla większych obrazów kolorowych) – o ile JPEG-LS był kilka razy szybszy niż PNG, znaczący przeskok zanotowano pomiędzy tymi 2 algorytmami a JPEG2000 – ten ostatni był nawet 10 razy wolniejszy niż JPEG-LS. Dla obrazach w odcieniach szarości różnice te były mniejsze (co jednak jest też spowodowane mniejszym rozmiarem grafik), jednak proporcje pozostały te same – najszybciej wykonywała się kompresja do formatu JPEG-LS, wyprzedzając PNG i JPEG2000.